Numerical analysis of a dual-sediment transport model applied to Lake Okeechobee, Florida

Stuart R. Clark, Wenjie Wei and Xing Cai

Computational Geosciences, CBC, Simula Research Laboratory,
Martin Linges vei 17, Fornebu, 1364, Norway
E-mail: {`stuart,xingca and wenjie`}`@simula.no`

Abstract

In this work, we study two numerical strategies for solving a coupled system of distinct nonlinear partial differential equations, which can be used to model dual-lithology sedimentation. Using high-resolution bathymetry data of Lake Okeechobee, Florida, we study the stability and computational speed of these numerical strategies. The fully-explicit scheme is straightforward to implement and requires a relatively small amount of computation per time step. However, this simple numerical strategy has to use small time steps to ensure stability. These small timesteps may render the explicit solver impractical for long-term and high-resolution basin simulations. As a comparison, we have implemented a semi-implicit scheme, where the two partial differential equations at each time step are solved implicitly in sequence. This semi-implicit scheme is numerically stable even for very large time steps. Using parallel computing, we have applied both schemes to a realistic case, Lake Okeechobee, Florida. The simulation successfully diffused material along a river-channel and into the lake. Both MPI-based implementations demonstrated satisfactory parallel efficiency on a multicore-based cluster.

1 Introduction

Processes of sediment erosion and deposition are often modelled using a diffusion equation with the regional slope and transport efficiency governing the speed of diffusion [7]. In models with different types of sediment, another equation can be added for each additional sediment type to account for the ratio of the sediment types [9]. A dual-lithology model for siliclastic sediment transport has formed a basis for some of the deterministic diffusion-based models of sediment transport existing today [2]. In addition to having an underlying diffusion equation, numerous empirical rules are incorporated that govern the different physical processes: sediment compaction, tectonic movements and carbonate production, for example. By neglecting the additional rule-based formulas, we can investigate the parallel performance of the dual lithology diffusion equation alone.

1.1 Lake Okeechobee

As a basis for testing the numerical properties of the model, we chose Lake Okeechobee, in southern Florida. Data for the lake's bathymetry was derived from depth-sounding at approximately 3 m intervals along north-south lines, each about 1000 m apart [4], while the lake itself is almost 60 km in diameter. We have chosen Lake Okeechobee precisely for this publicly available high resolution data, which allows us to test the numerical implementation with a large number of unknowns. To complete the square computational domain, a 30-arc second (approximately 1 km) grid, GTOPO30 [11] was used as the basis for the surrounding topography. The combined elevation map of the computational grid is shown in Figure 1. The lake is extremely shallow, on average 2.7 m deep [10] and at its deepest in only about 4 m of water. Kissimmee River is the largest single contributor of sediment to the northeast, accounting for 30 % of the water volume coming into the lake [8].

1.2 Mathematical Model

Diffusion has been used to model sediment transport in fluvial, on- and off-shore environments. The equations were developed first for the case of landscape evolution and fluvial transport of sediment, but have found application in the continental shelf primarily because of the work of Jordan and Flemmings [6], in which the basic equation is $\frac{\partial h}{\partial t} = \nabla \cdot (\kappa \nabla h)$. Here h is the height of the basin, t is time and κ is the efficiency of diffusion, the so-called transport coefficient. Rivenæs [9] later modified the equation to take

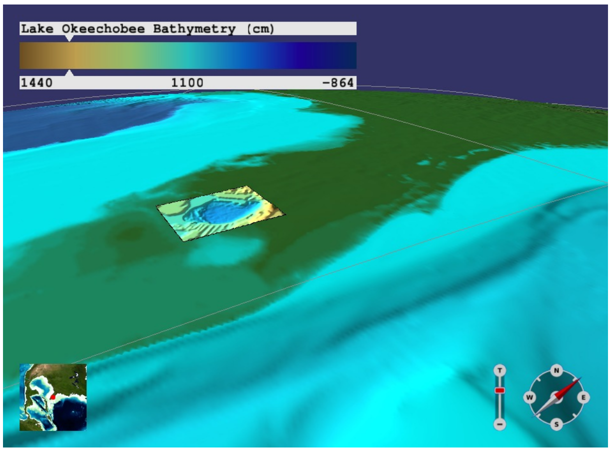

Figure 1. Florida viewed from the south-east. The lower-layer is exaggerated elevation from ETOPO2 [12] as veiwed from the southeast. Lake Okeechobee's present-day elevation, used in the computations, is displayed as the upper-layer.

into account two sediment types, each with their own transport coefficients. In the following equations, α and β are the two diffusion coefficients for each sediment type and s and $1 - s$ are the corresponding fractions of each sediment type at a particular location:

$$\frac{\partial h}{\partial t} = \nabla \cdot (\alpha s \nabla h) + \nabla \cdot (\beta(1-s)\nabla h) \qquad (1)$$

and

$$A\frac{\partial s}{\partial t} + s\frac{\partial h}{\partial t} = \nabla \cdot (\alpha s \nabla h). \qquad (2)$$

A is the layer thickness which scales the partial derivatives of s with respect to distance and determines how effective changes in h will be in affecting the sediment concentration. For our case, we set A equal to 1 m, representing the height of unsettled sediment in the basin. In our case, s represents first material type, sand, while $(1 - s)$ represents the second: silt.

The separation into two different material types, rather than having a diffusion coefficient based on grain-size, means that water discharge rates, water-flow induced shear-stress and drag-coefficients can be neglected [7]. For example, without a sediment source, the sediment type with the higher transport coefficient will move faster down-slope and preferentially fill deeper parts of the basin. In our case, we use coefficients for sand and silt, such that the fine-grain silt has a fractionally higher transport efficiency compared with sand (see Table 1).

Table 1. Transport coefficients by region in square metres per year (m^2/yr).

Region	Sand (α)	Silt (β)
Kissimmee River	70,000	100,000
Lake Okeechobee	2,100	3,000
Surrounds	70	100

The diffusion parameters for the lake region are modelled on Rivenæs' [9, p.64] except that, for simplicity, the values are not-depth dependent. We utilise lower values than those suggested by the lake's mean depth of 2.7 m, since we assume sediment diffusion is not as vigourous in the lake as for the seashore modelled in [9]. For the river, values an order of magnitude higher are chosen following Angevine et al. [1, cited in [9]]. Values two orders of magnitude lower than the lake are chosen for the surrounding flatlands, as we want the main sediment source into the lake to be from the river.

The numerical model is calculated using Neumann boundary conditions for the entire boundary, $\frac{\partial h}{\partial t}$ and $\frac{\partial s}{\partial t}$ are zero except along a 1.5 km wide channel, representing the River Kissimmee. Here, we set a Direchlet condition for s and non-zero Neumann for $\frac{\partial h}{\partial t}$. This allows us to specify inflow, for which we allow an 80% inflow of silt, 20% sand and $50m^2/yr$ per metre width of the channel or a total volume influx of 75,000 cubic metres per year.

The numerical model does not directly take into account wave action, thought to be of paramount importance in modelling sediment transport in Lake Okeechobee on short timescales such as months [5]. However, we are modelling the lake over a thousands of years, an appropriate scale for approximating wave-action by diffusion [2].

2 Two numerical methods

Computer basin simulations using the above mathematical model correspond to solving an initial-value problem by time integration. At time step ℓ, the latest numerical solutions of h and s, denoted by $h^{\ell-1}$ and $s^{\ell-1}$, are used as initial values. The time derivatives are approximated by

$$\frac{\partial h}{\partial t} \approx \frac{h^{\ell} - h^{\ell-1}}{\Delta t}, \quad \frac{\partial s}{\partial t} \approx \frac{s^{\ell} - s^{\ell-1}}{\Delta t}.$$

Different discretisations of the right-hand sides of (1)-(2) will give rise to different numerical strategies, which often have different characteristics in accuracy, stability and computational speed. In this paper, we consider two methods as follows.

2.1 A fully-explicit numerical scheme

If values of $h^{\ell-1}$ and $s^{\ell-1}$ are used in the spatial discretisation of the right-hand side of (1), while values of h^ℓ and $s^{\ell-1}$ are used for the right-hand side of (2), we can arrive at a fully-explicit numerical scheme:

$$\begin{aligned}\frac{h^\ell - h^{\ell-1}}{\Delta t} = \; & \nabla \cdot (\beta(1-s^{\ell-1})\nabla h^{\ell-1}) \\ & + \nabla \cdot (\alpha s^{\ell-1} \nabla h^{\ell-1}) \qquad (3)\end{aligned}$$

and

$$A\frac{s^\ell - s^{\ell-1}}{\Delta t} + s^\ell \frac{h^\ell - h^{\ell-1}}{\Delta t} = \nabla \cdot (\alpha s^{\ell-1}\nabla h^\ell). \qquad (4)$$

The diffusion terms with respect to $h^{\ell-1}$ on the right-hand side of (3) can be discretised by central finite differences. The convection term on the right-hand side of (4) with respect to $s^{\ell-1}$ can be discretised by upwind finite differences. The advantage of the fully explicit scheme (3)-(4) is that no linear systems need to be solved, so that the values of h^ℓ and s^ℓ on each mesh point can be computed by simple algebraic operations.

However, the disadvantage of the fully-explicit scheme is that the size of Δt is limited by the values of α, β, and mesh spacing. In particular, since (1) is a diffusion equation, the maximum allowed Δt for (3) is of order $\mathcal{O}(\Delta x^2/\max(\alpha, \beta))$. Such a limit may be too restrictive for practical long-term and high-resolution simulations.

2.2 A semi-implicit numerical scheme

As a remedy to the stability problem inherent to the fully-explicit scheme, we consider a so-called semi-implicit scheme as follows:

$$\begin{aligned}\frac{h^\ell - h^{\ell-1}}{\Delta t} = \; & \nabla \cdot (\beta(1-s^{\ell-1})\nabla h^{\ell}) \\ & + \nabla \cdot (\alpha s^{\ell-1} \nabla h^{\ell}) \qquad (5)\end{aligned}$$

and

$$A\frac{s^\ell - s^{\ell-1}}{\Delta t} + s^\ell \frac{h^\ell - h^{\ell-1}}{\Delta t} = \nabla \cdot (\alpha s^{\ell}\nabla h^\ell). \qquad (6)$$

The reason for terming the above scheme as semi-implicit is because (5) and (6) are both of an implicit and linearised form, but can be solved by two separate linear systems. This is unlike a fully-implicit scheme which requires finding all the h^ℓ and s^ℓ values simultaneously by solving a larged coupled system of nonlinear algebraic equations.

2.3 Parallel implementation

We have adopted a straightforward parallelisation approach, where the global solution domain is divided into rectangular subdomains, each assigned to one processor core. MPI has been used for the inter-subdomain communicaiton. For the semi-implicit scheme, we have used the parallel iterative CG and GMRES solvers from the Trilinos package [3].

3 Numerical experiments

In this section we will first examine a comparison of two numerical strategies using stability and computational speed as criteria. This investigation is done using a realistic basin elevation as the basis for the simulation, Lake Okeechobee, where the spatial domain spans a rectangular area of 62.13 × 69.50 km. The initial topography of the basin, i.e., $h(x, y, 0)$, is depicted in Figure 2, while the initial sand-silt volume fraction is assumed to be a constant at 50%, that is $s(x, y, 0) = 0.5$. Other parameters are as per Section 1.2. Afterwards, we present some speedup measurements of the two MPI-based parallel implementations. Finally, we discuss the results of a 10,000 year simulation.

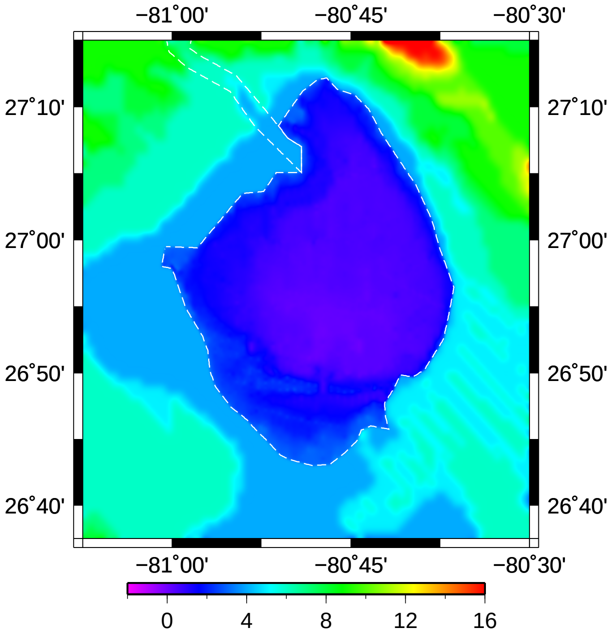

Figure 2. Inital height, h, in meters. Outlines of Lake Okeechobee and the Kissimmee River used to define the transport coefficients are dotted in white.

All the numerical experiments have been carried out on a cluster of multicore-based compute nodes, where each node

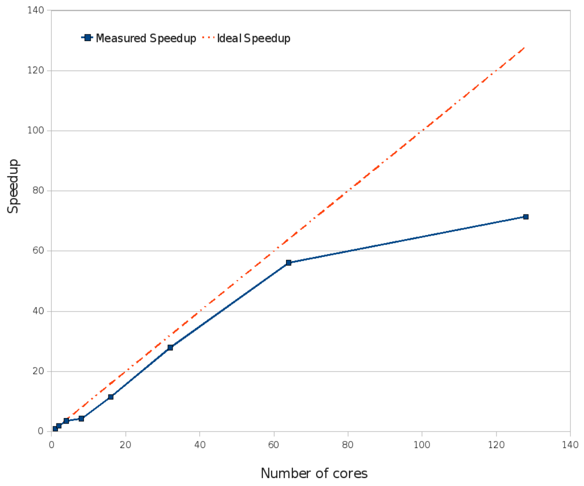

Figure 3. Speedup results of the fully-explicit scheme.

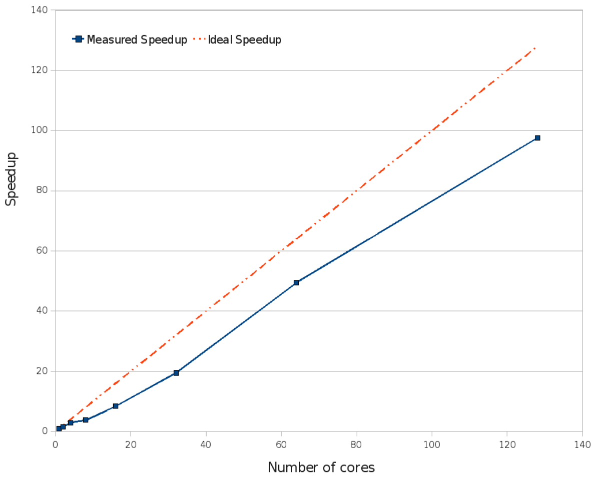

Figure 4. Speedup results of the semi-implicit scheme.

is equipped with two Xeon 2.66 GHz quad-core processors, i.e., 8 cores per node. The interconnect network between the nodes is InfiniBand.

3.1 Stability

For the purpose of checking the stability requirement of the fully-explicit scheme, we have used two spatial mesh resolutions: 250×250 and 1000×1000. On the coarser mesh, the maximum allowed Δt for the fully-explicit scheme is approximately 0.21 year, whereas the maximum allowed Δt is approximately 0.013 year for the finer mesh. For the semi-implicit scheme, we tried Δt of order of thousands of years on both meshes, and did not encounter any stability problem.

3.2 Computational speed and speedup

Studying the computational speed of the two MPI-based implementations, we used a 10-year simulation on the 1000×1000 mesh, while using $\Delta t = 0.01$ year for both methods. Table 2 and Figures 3 and 4 show the actual time measurements and speedup results. From Table 2, it can be seen that the semi-implicit approach is approximately a factor of 100 slower calculating each time step. This large difference in the computational speed is due to two reasons. Firstly, the semi-implicit scheme is quite complex, where two linear systems have to be set up and solved separately per time step. We have used the Conjugate Gradient iterative method for solving the linear system arising from (5) and the GMRES iterative method in connection with (6). And secondly, the MPI-implementation of the semi-implicit scheme has not undergone extensive code optimisations as those that have been applied to the fully-explicit implementation.

Table 2. Time measurements (in seconds) of the two numerical schemes for a 10-year simulation on the 1000×1000 mesh, $\Delta t = 0.01$ year.

Number of cores	Fully-explicit	Semi-implicit
1	55.00	5102.13
2	29,46	3230.01
4	15.62	1748.45
8	12.84	1349.22
16	4.77	608.65
32	1.97	262.56
64	0.98	103.08
128	0.77	52.30

The fully-explicit scheme has better parallel efficiency when the number of cores used is small. On the other hand, the semi-implicit scheme scales better on a large numbers of cores. This is clearly due to the unfavourable computation-communication ratio of the fully-explicit scheme when the 1000×1000 mesh is divided into many subdomains. However, as Figures 3 and 4 demonstrate, both MPI-implementations have obtained reasonable speedups on the multicore-based cluster.

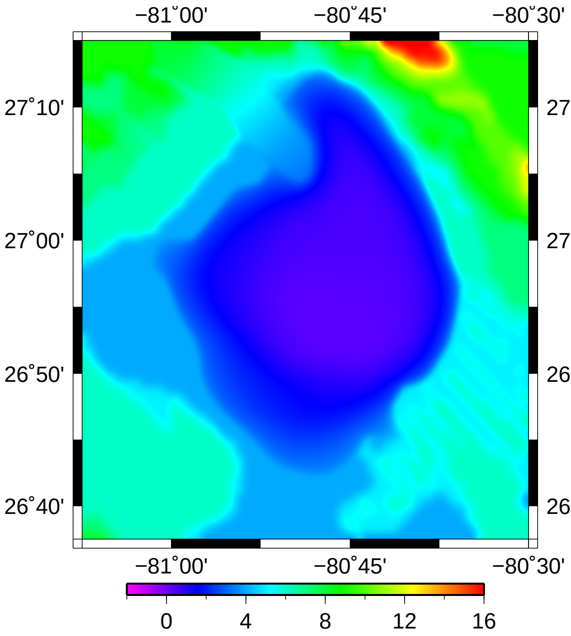

Figure 5. h **after 1,000 years.**

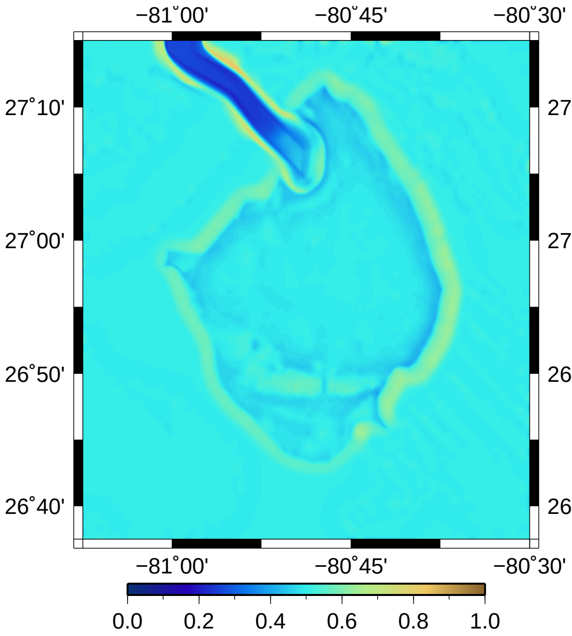

Figure 6. s **after 1,000 years.**

3.3 Diffusion-based evolution of Lake Okeechobee over 10,000 years

Using both MPI-implementations, where the fully-explicit method uses $\Delta t = 0.01$ year and the semi-explicit method uses $\Delta t = 0.1$ year, we carried out a 10000-year simulation; the two numerical methods produced qualitatively similar simulation results.

Figures 5-6 show the evolution of elevation and sand-content in and around Lake Okeechobee after 1000 years of model evolution. In general, silt moves to the lowest lying sections of the basin, leaving sand exposed at the lake's edge. The same applies to the channel, leaving a sandy bank exposed on either side of the river. The three order of magnitude contrast between the channel and the flatlands inhibits diffusion out of the channel, while material diffuses perferentially along the river, leading to fan-shaped bank of sediment at the river mouth, where the transport coefficients are lower by an order of magnitude.

Allowing the model to evolve further, until 10,000 years, the basin bathymetry and surrounding topography are sharply smoothed by the diffusion equation. On the other hand, the sand fraction has a large high-frequency signal and, at this point, correlates with the initial topography well: silt predominately occupies previously lower elevated areas, while sand is left in areas with initially high elevation, except in the region of the river. In addition, the injection of mud around the river mouth is now highly pronounced.

4 Conclusion

Using a purely diffusion-based model, we have simulated inflow of material along River Kissimmee and into Lake Okeechobee. Of the two schemes adopted to solve the model equations, the semi-implicit scheme had superior stability in all of our numerical experiments. The current algorithm uses backward Euler discretisation in time, which gives first-order temporal accuracy: the same as the fully-explicit scheme. In the future, we can apply the Crank-Nicholson method in the time direction of the semi-implicit scheme, potentially obtaining second-order temporal accuracy. Further code optimisations can also be carried out, with the aim of reducing the CPU time consumption. As a result of the above, we expect the semi-implicit scheme to be of better practical use when higher spatial resolutions force the fully-explicit scheme to adopt prohibitively small time steps. These schemes could also be applied to cases of sediment transport into paleo-basins and, given we expand the model, used to predict the geology of the basin. The expansion of the model to include, for example, sediment compaction and tectonics, may pose additional challenges to the stability and parallel efficiency of the schemes.

Acknowledgements

This work was partially funded by a research grant from Statoil. The authors thank Statoil and Kalkulo for the use of the *4D Lithosphere Model*, used to prepare Figure 1. The research has been conducted at Simula Research Laboratory

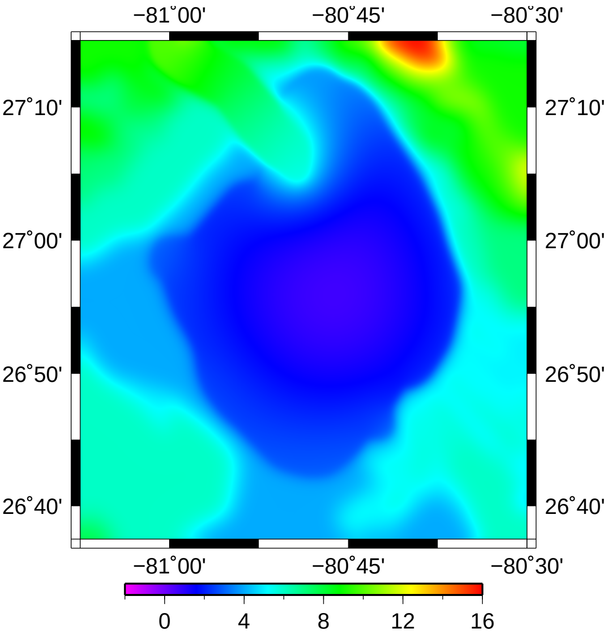

Figure 7. h **after 10,000 years.**

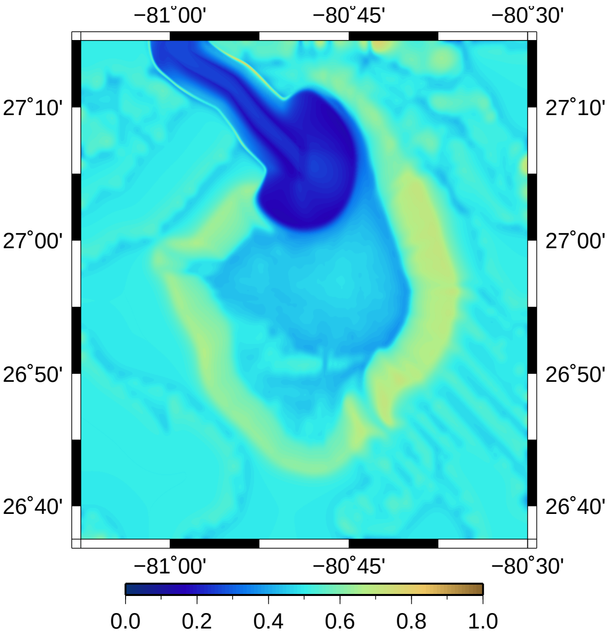

Figure 8. s **after 10,000 years.**

as part of the CBC, a Center of Excellence funded by the Research Council of Norway.

References

[1] C. L. Angevine, P. L. Heller, and C. Paolo. *Quantitive sedimentary basin modeling*. Continuing Education Course Note Series. The American Association of Petroleum Geologists, Tulsa, 1990.

[2] D. Gradjeon and P. Joseph. Concepts and applications of a 3-d multiscale lithology diffusive model in stratigraphic modeling. In W. Harbaugh, L. Watney, C. Rankey, R. Slingerland, R. Goldstein, and E. Franseen, editors, *Numerical Experiments in Stratigraphy: Recent Advances in Stratigraphic and Sedimentologic Computer Simulations*, volume SEPM Special Publications, pages 197–210. Society for Sedimentary Geology, 1999.

[3] M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams. An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

[4] G. W. Hill, N. T. DeWitt, and M. Hansen. Lake okeechobee bathymetry data. Technical report, U.S. Geological Survey, 2002. http://sofia.usgs.gov/publications/maps/lakeokeebathy/index.html.

[5] K. R. Jin and Z. G. Ji. Case study: Modeling of sediment transport and wind-wave impact in lake okichobee. *Journal of Hydrological Engineering*, 130:1055–1067, 2004.

[6] T. E. Jordan and P. B. Flemmings. Large-scale stratigraphic architecture, eustatic variation, and unsteady tectonism: a theorectical evaluation. *Journal of Geophysical Research*, 96:6681–6699, 1991.

[7] C. Paola. Quantitive models of sedimentary basin filling. *Sedimentology*, 47:121–178, 2000.

[8] K. R. Reddy, O. A. Diaz, L. J. Scinto, and M. Agami. Phosphorus dynamics in selected wetlands and streams of the lake okeechobee basin. *Ecological Engineering*, 5:183–207, 1995.

[9] J. C. Rivenæs. *A computer simulation model for siliclastic basin stratigraphy*. PhD thesis, University of Trondheim, 1993.

[10] S. P. Schottler and D. R. Engstrom. A chronological assessment of lake okeechobee (florida) sediments using multiple dating markers. *Journal of Paleolimnology*, 36:19–36, 2006.

[11] U. G. Survey. GTOPO30, 1996.

[12] N. O. U.S. Department of Commerce and N. G. D. C. Atmospheric Administration. 2-minute gridded global relief data (etopo2v2), 2006.